INFLUENCE OF COMPUTER COMPUTATION PRECISION IN CHAOS ANALYSIS

by Valentin STEFANESCU[1], Dan STOICHESCU[2], Madalin FRUNZETE[3] and Bogdan FLOREA[4]

Această lucrare işi propune realizarea unui studiu al influenţei preciziei de calcul asupra comportamentului haotic al unui circuit. Acest tip de haos, cauzat de precizia de calcul, va fi denumit generic haos computaţional pe parcursul lucrării. Studiul se efectuează pe baza unui oscilator Alpazur cu 2 stări iar rezultatele vor fi interpretate din punct de vedere al exactităţii lor dar şi a duratei necesare pentru obţinere. Se vor trage concluzii pe baza timpilor de rulare şi pe baza diferenţelor între rezultatele simulării şi cazul ideal. De asemenea este propusă şi o metodă de cuantificare a haosului şi de asemenea şi de determinare a preciziei necesare pentru obţinerea de rezultate considerate satisfăcătoare.

In this paper the influence of computer simulation on a chaotic circuit behavior is studied. This type of chaos determined by the computer accuracy is called computational chaos (or computer chaos) throughout this paper. A two state Alpazur oscillator is used in the study and the results are interpreted precisionwise and timewise (time needed to run one simulation). The conclusions are drawn on the basis of necessary time to run a simulation; the analytical and experimental results are compared. A chaos quantification method used in determining the necessary precision for getting good results is proposed.

Keywords: alpazur, chaos, computer simulation, precision comparison

1. Introduction

This paper presents some effects of computer simulations on chaotic behaviour of hybrid dynamical circuits. It is known from literature [1] that research needs several steps. The first step consists in performing a very solid

[1]Department of Applied Electronics and Informations Engineering, University POLITEHNICA of Bucharest and ENSEA, Universite de Cergy Pontoise, Paris, France, e-mail:valentinstefanescu@gmail.com

[2]Prof., Department of Applied Electronics and Informations Engineering, University POLITEHNICA of Bucharest, e-mail: stoich@elia.pub.ro

[3]Department of Applied Electronics and Informations Engineering, University POLITEHNICA of Bucharest, e-mail: madalin.frunzete@upb.ro

[4]Department of Applied Electronics and Informations Engineering, University POLITEHNICA of Bucharest, e-mail: bogdan.florea@upb.ro

theoretical study [2, 3]. This gives a better understanding of what is to be achieved. The second step is some form of application that will give better insight on the studied phenomena. This can be either an experimental step or an intermediary computer simulation step. In modern studies, the cases when going from theoretical study straight to the experimental stage are less and less frequent. Almost all analysis include a computer modeling and simulation step.

Actual experiments are very important but sometimes difficult to perform. They may be very expensive and can lead to disastrous outcomes. Computer simulations, on the other hand, are relatively easy to perform. Generally, a combination of computing power and software to harness the respective computing power is sufficient.

In the second chapter of this paper, a research based on an Alpazur oscillator chaotic behavior is presented. It contains all the details related to the used computational environment (hardware, software, time necessary to run a simulation).

The third chapter is focused on the research results. Relevant as well as less relevant results are examined and explained. There is, also, an in depth analysis of the relationship between result precision and usefulness (the paper attempts to provide a guideline to any researcher who would obtain useful results for a similar application in a minimum amount of time).

The final chapter presents further possible usefulness of the paper results; it shows also the benefits of parallel computing in simulations.

2. Problem Statement

In order to better explain what the study in this paper refers to, a short presentation of the used application is necessary. The application was firstly developed by Kawakami and Lozi, is called Alpazur oscillator [4] and presents an unique set of features: it is not just a simple nonlinear application, but a hybrid dynamical system as it presents both a nonlinear continuous component as well as a discontinuity. This type of system is very efficient in determining the ability of the computer to simulate such a behaviour due to its mixed nature. The circuit exhibits chaotic behavior too. Chaos in general is very sensitive to the experimental environment (real or virtual) [5–7] and it can be used as a level trigger to determine when the simulation has lost its usefulness.

So, for this model, the computation precision is varied. It will cover a wide range of precision variation (from a very precise simulation requiring a long amount of time to very fast simulations with unreliable results) and also will allow better setup of the simulation, meaning that the circuit component values are easily adjustable to obtain stability or chaos in the analytical case.

2.1. The Alpazur oscillator. This circuit is a 2-state hybrid dynamical system. The two states are realized via a set of two power sources and a switch. The switch introduces the discontinuity in an otherwise simple oscillator. A thorough study of this circuit was performed by Quentin Brandon [8], study that covered several types of Alpazur oscillators (the two-state, the three-state and the 3D versions). It is very interesting that this circuit exhibits chaotic behaviour in precise given conditions with only two states and no delays. The system is shown in Fig. 1.

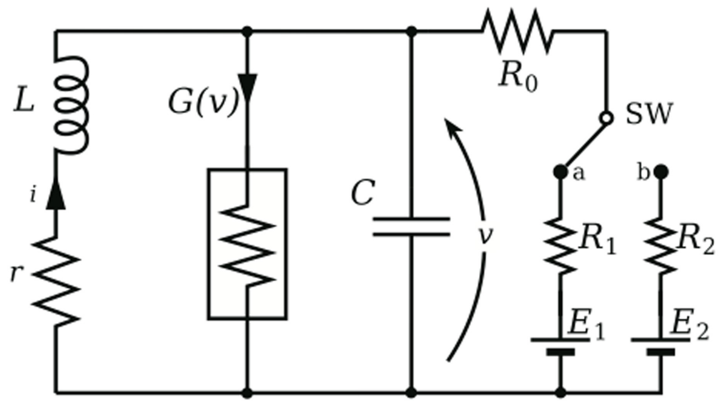


Fig. 1. 2-state alpazur oscilator circuit

The circuit is described by the equation (1):

$$\begin{cases} L\frac{di}{dt} = -ri - v \\ C\frac{dv}{dt} = i - g(v) + \frac{E_j - v}{R_0 + R_j} \end{cases} \quad \text{where j=1, 2} \tag{1}$$

In eqs. 1 the values of i and v are marked in Fig. 1.

$$g(v) = -a_1 v + a_3 v^3, \text{ where } a_1, a_3 > 0 \tag{2}$$

The main interest is to properly determine i and v as the parameters that define the state of the system at a given time. Therefore the following notations and changed variables define the state of the system:

$$X = \begin{pmatrix} x \\ y \end{pmatrix}, \text{ where } x = i\sqrt{L} \text{ and } y = v\sqrt{C} \tag{3}$$

Other necessary notations:

$$\tau = \frac{t}{\sqrt{LC}} \qquad r_n = r\sqrt{\frac{C}{L}} \qquad b = a_1\sqrt{\frac{L}{C}}$$
$$c = \frac{3a_3}{C}\sqrt{\frac{L}{C}} \quad A_j = \sqrt{\frac{L}{C}}\frac{1}{R_0 + R_j} \quad B_j = \sqrt{L}\frac{E_j}{R_0 + R_j}$$

To further simplify the matter, $a_1 = 1$ and $a_3 = -1/3$ are considered. After properly processing equation (1), the following relations are obtained:

$$\begin{cases} \frac{dx}{d\tau} = -r_n x - y \\ \frac{dy}{d\tau} = x + (1 - A_i)y - \frac{1}{3}y^3 + B_i \end{cases} \quad \text{where } i = 1, 2 \tag{4}$$

The discontinuity (switch) puts the system in two different states as one can see in Fig. 2.

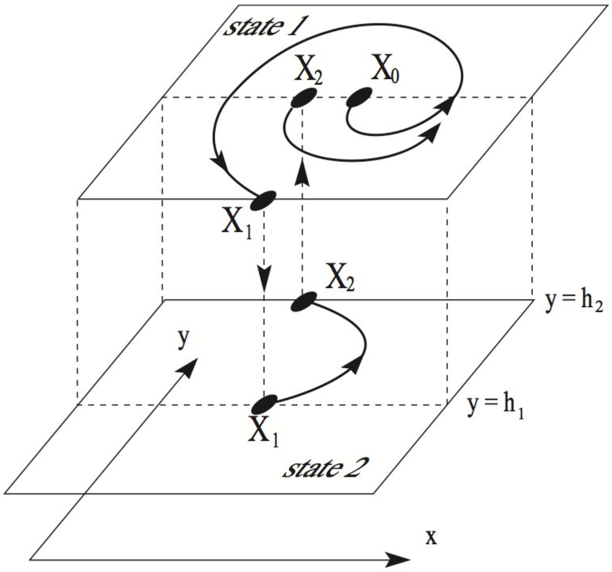


Fig. 2. System state based on selected source

The power sources have been replaced, for the sake of closer resemblance to an actual model, by square signal generators. The square signals are centred on E_1 and respectively E_2 and are oscillating between 0 and $2E_1$ respectively $2E_2$. This behavior makes the generators have similar functionality to a chopper.

$$\begin{array}{l} E_{j(generated)} = E_j + v_{square} \\ v_{square} = E_i \frac{4}{\pi} \sum_{k=0}^{\infty} \frac{\sin(2k-1)2\pi f t}{2k-1} \end{array} \text{, where j=1,2} \quad (5)$$

When the square signal frequency f is high, the output seems to be constant.

Using $E_i + v_{square}$ instead of E_i is very important since the paper studies computer influence over chaotic behaviour. A constant voltage generator would have eliminated a possible source of chaos.

2.2. **Chaos in the Alpazur.** As we can easily see, the Alpazur is a hybrid dynamical system [9] . Such a circuit can exhibit various types of behavior in terms of its initial values. It can be a stable circuit, characterized by one path present in the $i(v)$ plot. If some modification is done to the values of the load or power source, the system can be sent either in a chaotic state or a semi-chaotic state. Such a state will be called period doubling (or in some rare case period tripling) when the path described by the $i(v)$ plot presents two loops (or three). If the plot contains more than three loops, the case is considered chaotic. All these semi-chaotic states are very rare and can easily turn into either chaos or stability with very little influence from the outside. Such states are also influenced by the computational environment and selected precision.

3. Measurements and results

In order to be able to measure the degree of chaos in a $i(v)$ plot a method of quantification is necessary. There are several methods of quantification in literature [10–12], mostly based on Lyapunov exponents. These methods are applied to either discrete or continuous systems. In this case, the system is mixed (continuous with a discontinuity) and so none of the existing methods are usable. In order to define the necessary parameters to compute the degree of chaos, the $i(v)$ plot is used. When the commutation takes place at the same point on the $i(v)$ plot, the system is considered stable. If the points do not coincide on the $i(v)$ plot then the system is considered chaotic and the horizontal dispersion of the commutation points is used to quantify chaos. A similar method can be used to produce a representation called route to chaos [13–15] for a hybrid dynamical system. The route to chaos is used to analyse the change in behavior for a given circuit when a parameter is varied.

3.1. Measurement method. In order to quantify the chaotic behavior some initial parameters have to be defined: N and ϵ.

N represents the number of points that will be used (this is constant and is set by the period of time set for the simulation to run and also by the frequency of the oscillator).

ϵ represents the allowed vicinity. This parameter will allow two close points to be considered distinct or identical.

If there are N points used:

$$N = n_1, n_2, n_3... \tag{6}$$

in order for n_1 to be different from n_2, the following rule has to be applied:

$$n_1 + \epsilon < n_2 \; or \; n_1 - \epsilon > n_2 \tag{7}$$

So far, this allows us to distinguish the points one of each other and detect the overlapped points (considered identical). This also allows the detection of stable circuits. A circuit is considered stable if:

$$n_k - n_l < \epsilon, \; \forall n_k, n_l \in N \tag{8}$$

In order to compare chaoswise two simulations, the method of quantification needs to include both the dispersion and the distribution of the points.

Dispersion is defined as the distance between the lowest point and the highest point:

$$D = min(n_k) - max(n_k), \text{ where } n_k \; \epsilon \; N$$

Distribution is defined as the maximum distance between two adjacent points

$$d = max(n_k - n_l), \text{ where } n_k, n_l \epsilon N \quad (9)$$

and

$$(\nexists)\ n_j \epsilon N, \text{where } (n_k - n_j) + (n_j - n_l) = n_k - n_l \quad (10)$$

The distance between two points is considered to be difference of their x-axis value.

$$dist(n_k, n_l) = |x_{n_k} - x_{n_l}| = |i(n_k) - i(n_l)| \quad (11)$$

The x-axis values considered are actually the values of the current i at which the switch takes place. The difference between the two is always considered positive.

In order to have a proper characterization of the point pattern we will need to define density as well:

$$\rho = \frac{\sum dist(n_k, n_k+1)}{(N-1)}$$

where

$$(\nexists) n_i | dist(n_k, n_i) + dist(n_i, n_k + 1) = dist(n_k, n_k + 1)$$

Density represents the average distance between two adjacent points.

In order to draw a proper conclusion it is needed to establish a method to relate the components defined above and compute the degree of chaos:

$$C_h = Dd\rho$$

This rather simple method allows an objective analysis of two different sets of results. Since this is a computer simulated experiment and the comparison between results is also done by a computer, the results may be altered by the method but not in a manner that will prevent reasonable analysis.

3.2. **Simulation and results.** The simulation environment was based on Matlab 2011a version using a Simulink .mdl file to model the circuit and Matlab scripting for results post processing.

Since these simulations regard computer precision, we chose a constant step for the .mdl file simulation. The step is defined as a fragment of a second. The simulation starts with precisions of $10^{-3}s$ to $10^{-8}s$. This evolution can be applied to different situations. For example, in this case, the source is considered a voltage constant one, but can be changed into a square signal generator of a given frequency. For now, the simulation is performed with a single set of initial values that allows a stable initial state (as confirmed by mathematical analysis).

In order to define the stability of the circuit, a specific set of values was selected:

$$E_1 = 147V,\ E_2 = -100V,\ R_1 = 460\Omega\ R_2 = 10\Omega$$
$$r = 10\Omega\ R = 40\Omega,\ L = 1mH,\ C = 1mF.$$

These values position the Alpazur oscillator approximately at 160Hz. Of course, this is not a fast oscillator but it is good enough to give an idea of the precision needed. This means that for every increase in frequency, the precision needs to be adjusted accordingly.

An ideal result for a simulation in this point would look similar to Fig. 3

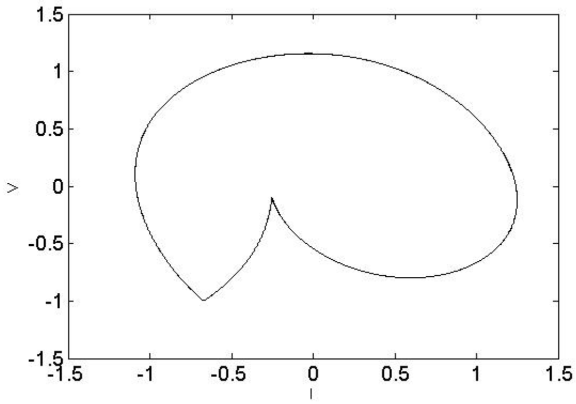


Fig. 3. Ideal simulation

This means that the simulation results in a single loop that ends in the exact same place it has started.

The first simulation will use a 10^{-3} precision step. The result is visible in Fig. 4

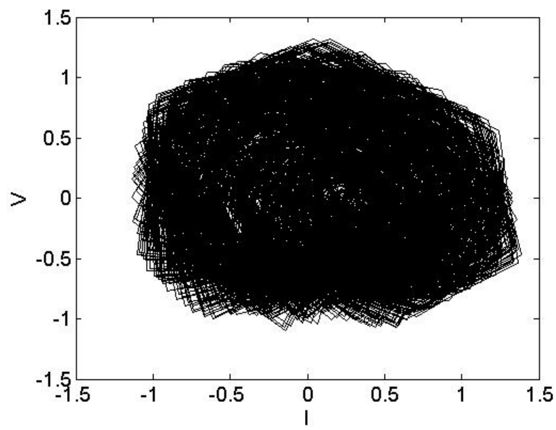


Fig. 4. Simulation result with 10^{-3} precision

One can easily see that the circuit appears to be very chaotic so this is not a very good measure of precision to be used in order to obtain proper values in a computer simulation. The degree of chaos computed as described earlier is:

$$C_h = Dd\rho = 1.4580$$

In order to estimate the meaning of such a value, it must be mentioned that a value above 0.1 corresponds to a severely chaotic circuit and a value of 0 corresponds to a perfectly stable one. Unfortunately 0 cannot be obtained.

The dependence of the chaos degree on the precision has to be interpreted from the point of view of a multiplication factor and not of its absolute value.

The chaos degree may be expressed as:

$$C_h = m10^n$$

Increasing the precision, the chaos degree decreases but, at a certain point, only n is modified.

For instance, if the precision is set to 10^{-4}:

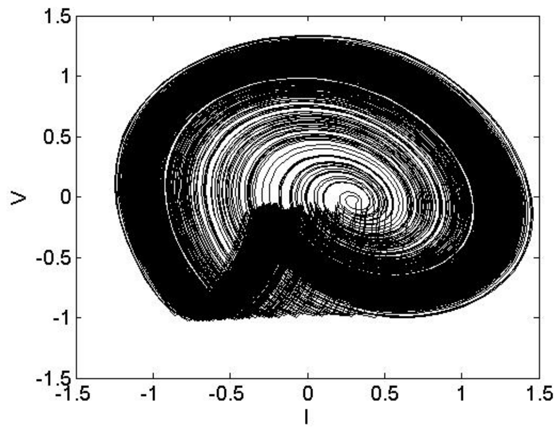


Fig. 5. Simulation result with 10^{-4} precision

the chaos degree measured value is: $C_h = 0.0418$ with $m = 4.180$ and $n = -2$

For 10^{-5} (Fig. 6), $C_h = 3.5148 \cdot 10^{-4}$ with $m = 3.514$ and $n = -4$

After 10^{-6} precision, the value of m appears to be stable as shown in Fig. 7 with $C_h = 3.5241 \cdot 10^{-6}$ with $m = 3.524$ and $n = -6$

For the rest of the simulations, the trend is maintained (Fig. 8).

For 10^{-7} precision, $C_h = 3.5651 \cdot 10^{-8}$.

For 10^{-8} precision, $C_h = 3.5656 \cdot 10^{-10}$.

The difference in the order of magnitude n is obtained due to the method of computation of C_h. One could deduce that once m appears to be constant, the computation precision of the machine has been reached and there is no need for further increase in precision.

Everything so far has been studied for a given frequency of the oscillator. The frequency used was roughly:

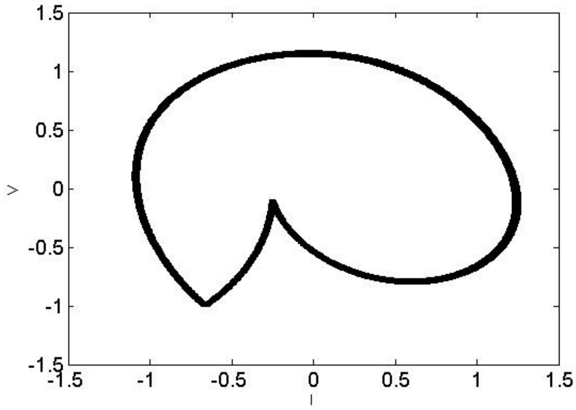


Fig. 6. Simulation result with 10^{-5} precision

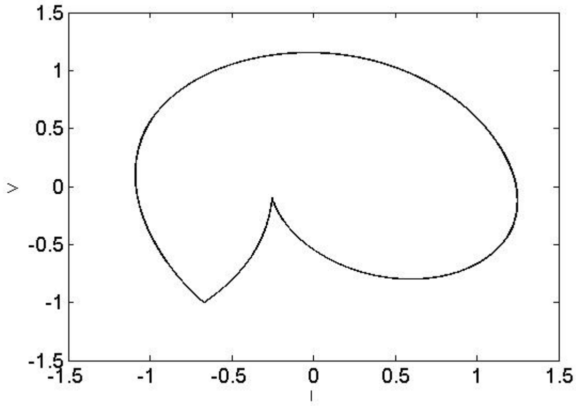
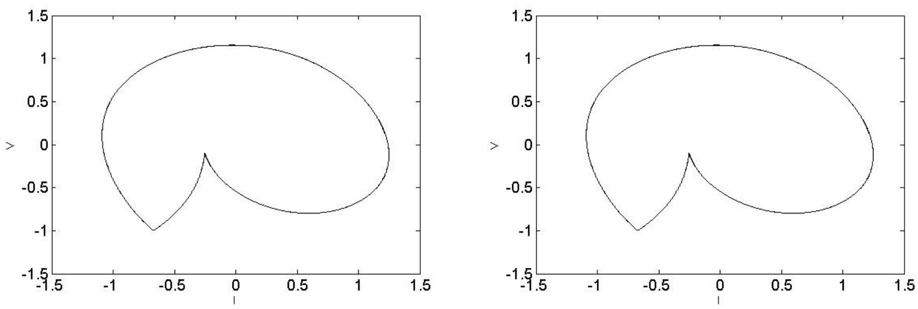


Fig. 7. Simulation result with 10^{-6} precision



(a) $10^{-}7$ precision

(b) $10^{-}8$ precision

Fig. 8. Simulation results for precisions up to $10^{-}8$

$$f = \frac{1}{2\pi\sqrt{LC}} = 160Hz \tag{12}$$

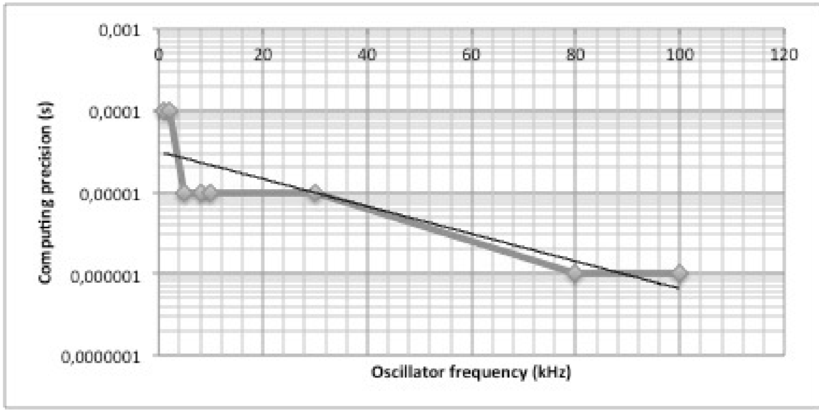


Fig. 9. Oscillator Frequency - Simulation Precision dependency

This is a low frequency for any modern electronic device. If all tests are to be repeated but the values of L and C are modified with an order of magnitude - $L = 100\mu H$ and $C = 100\mu F$ - the resulting frequency according to 12 becomes: $f = 1.6KHz$. This will appear as chaotic behavior if simulated with a precision of 10^{-7}. Raising precision to 10^{-8} the circuit appears to be stable.

4. Conclusions

The aim of this article is to point out the chaos coming out from other sources different from the already studied ones. Chaos, due to its nature, occurs in the most unexpected ways. Since most chaos studies are performed via computer simulations, the ideea of computational chaos represents an important factor as far as computer studies go on. This type of chaotic behaviour appears not only in chaotic circuits but, also in stable ones. So, it may be concluded that computational chaos cannot be eliminated but can be kept under control.

One can draw the conclusion that for oscillator frequencies around 100KHz, a step of 10^{-6} is a minimal requirement for good results. 10^{-7} is still better. Also, an increase of 100 times of the oscillator frequency will be covered well by a 100 times increase in precision. Consequently one could draw the conclusion that there is a linear dependence between the circuit frequency and the simulation precision.

According to several tests, the graph in Fig.9 shows the dependence between oscillator frequency and system precision in several situations. This set of results led to the assumption that the dependence is quite linear. The linearity reference here is made according to the amount of increase in precision compared to the amount of increase in oscillator frequency.

Also, for the actual given example, a frequency that requires a precision larger than 10^{-9} will require a very long simulation time; it is better to decrease frequency and keep total simulation time lower.

So, as a rough rule, for precision setting in this types of simulations, the precision can be:

$$s = \frac{1}{f}10^{-4} \tag{13}$$

These results are valid up to a value of 10^{-9} used for simulation precision. An increase in computing power that would allow for faster and higher precision simulations may provide extra information in the evolution of the quality of the obtained results. Such results will probably refine the general rule given by the equation 13 by adding additional parameters.

The second conclusion of this article is that independently of the available computational power, there is a certain limit of the precision for every given simulation from where any increase in precision is useless from the point of view of the results.

One could easily object, due to the fact that computers are getting faster every day and they allow more and more precise computations, there might not be a reason to worry. This is not always true. In the case of certain types of simulations, increasing precision beyond the necessary level may not affect the simulation process, but in the case study presented in this paper, the time necessary for each simulation grew exponentially. For a machine equipped with a 2.4GHz CPU and 4GB RAM memory, the time necessary for one single simulation grew up from under one second for precisions of 10^{-3} and 10^{-4} to 40 min for 10^{-7} and $23h$ for 10^{-8}.

The necessary time to get satisfactory results may vary if the technology for modelling the simulations is changed (for example if C++ or even assembler are used instead of Matlab) and the time may even be reduced by an order of magnitude, the evolution of necessary time to run a simulation is the same. The representation of the results is also limited by the actual resolution of the ploting equipment. This can easily be noticed by studying the two representations for 10^{-7} and 10^{-8}.

As a further research in this field, a more enhanced method that can allow precision selection on various types of circuits, not only LC type oscillators, could be developed . Also, the study can be extended to various types of control applied to a given system. If a control is digital and ideal, in theory there is no influence, but, if an analogical type of control is simulated, the precision should affect the behavior of the control mechanism.

Acknowledgement

The work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POS-DRU/88/1.5/S/61178.

REFERENCES

[1] M. A. Aziz-Alaoui, A. D. Fedorenko, R. Lozi, and A. N. Sharkovsky. Recovering trajectories of chaotic piecewise linear dynamical systems. In *Proc. Conf. 1st Int Control of Oscillations and Chaos*, volume 2, pages 230–233, 1997.

[2] A. Banai and F. Farzaneh. Theoretical investigation of the stability of the modes in an array of coupled oscillators for linear and circular arrangements. In *Proc. European Conf. Circuit Theory and Design*, volume 2, 2005.

[3] Masahiro Asada. Theoretical analysis of spectral linewidth of terahertz oscillators using resonant tunneling diodes and their coupled arrays. *Journal of Applied Physics*, 108(3):034504, 2010.

[4] H. Kawakami. Bifurcation of periodic responses in forced dynamic nonlinear circuits: Computation of bifurcation values of the system parameters. 31(3):248–260, 1984.

[5] N. M. Goubareni. Simulation of chaotic iterative processes in speed-independent computing networks. In *Proc. Fourth Euromicro Workshop Parallel and Distributed Processing PDP '96*, pages 27–32, 1996.

[6] A. S. Demirkol, V. Tavas, S. Ozoguz, and A. Toker. High frequency chaos oscillators with applications. In *Proc. 18th European Conf. Circuit Theory and Design ECCTD 2007*, pages 1026–1029, 2007.

[7] I. Dalkiran, F. Y. Dalkiran, K. Danisman, and R. Kilic. Prediction of dynamics of chaotic chua's circuit with artificial neural network. In *Proc. IEEE 15th Signal Processing and Communications Applications SIU 2007*, pages 1–4, 2007.

[8] Quentin BRANDON. Numerical method of bifurcation analysis for piecewise-smooth nonlinear dynamical systems. 2009.

[9] A. Benveniste and P. Le Guernic. Hybrid dynamical systems theory and the signal language. 35(5):535–546, 1990.

[10] F. Rauf and H. M. Ahmed. Calculation of lyapunov exponents through nonlinear adaptive filters. In *Proc. IEEE Int Circuits and Systems Sympoisum*, pages 568–571, 1991.

[11] C. A. Pickover. Visualizing chaos: Lyapunov surfaces and volumes. 10(2):15–19, 1990.

[12] Fei Gao, Yibo Qi, Qiang Yin, and Jiaqing Xiao. A novel non-lyapunov approach in discrete chaos system with rational fraction control by artificial bee colony algorithm. In *Proc. IEEE Int Progress in Informatics and Computing (PIC) Conf*, volume 1, pages 317–320, 2010.

[13] P. S. Bodger, G. D. Irwin, D. A. Woodford, and A. M. Gole. Bifurcation route to chaos for a ferroresonant circuit using an electromagnetic transients program. *IEE Proceedings-Generation, Transmission and Distribution*, 143(3):238–242, 1996.

[14] S. Basu, S. A. Maas, and T. Itoh. Quasi-periodic route to chaos in a microwave doubler. 5(7):224–226, 1995.

[15] D. Arroya-Almanza, A. Pisarchik, and F. Ruiz-Oliveras. Route to chaos in a ring of three unidirectionally coupled semiconductor lasers. (99), 2012. Early Access.